AMAST'02 - St. Gilles les Bains

On Solving Temporal Logic Queries*

*with 10 figures

Samuel Hornus ∧ Philippe Schnoebelen

iMAGIS (lab. GRAVIR/IMAG - proj. CNRS / INPG / INRIA / UJF)

LSV (ENS Cachan / CNRS)

Overview

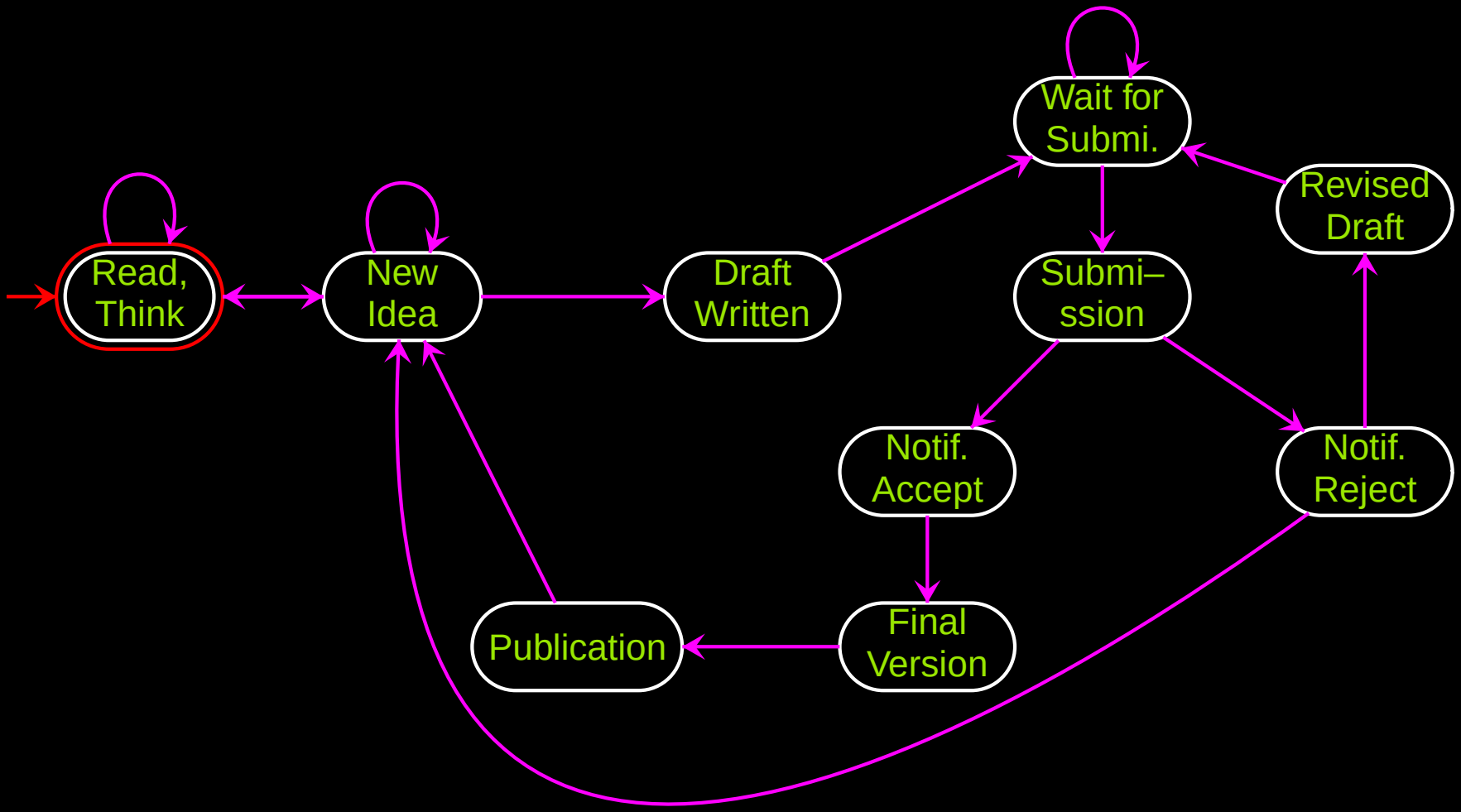
- Temporal logic queries
- The lattice of solutions
- Polynomial time algorithms
- Hardness results
- Conclusion

Overview: temporal logic queries (T.l.q.)

- Temporal logic queries
- The lattice of solutions
- Polynomial time algorithms
- Hardness results
- Conclusion

T.l.q.: kripke structure

A Kripke structure is a tuple $\langle \mathcal{Q}, q_{init}, \rightarrow, l \rangle$



T.l.q.: run

Let $\mathcal{S}$ be a Kripke structure.

$q_0 \to q_1 \to q_2 \to \ldots$ *is a run* π *of* $\mathcal{S}$*:*

$(q_i, q_{i+1}) \in \to$ *for all* i*.*

We write $\Pi(q)$ *for the set of runs that start from* q*.*

T.l.q.: temporal logic formula

A temporal logic formula φ expresses properties of a run π in a Kripke structure.

$$\pi \models \varphi$$

T.l.q.: model checking

Input: $(\mathcal{S},\varphi)$

$\mathcal{S} \models \varphi$ means $\forall \pi \in \Pi(q_{init}), \pi \models \varphi$

Output: $\mathcal{S} \models \varphi$?: does $\mathcal{S}$ satisfy φ ?

What if $\mathcal{S} \not\models \varphi$?

T.l.q.: temporal logic queries...

Let $\varphi = \mathsf{G}(idea \Rightarrow \mathsf{F}publication)$

Suppose $\mathcal{S} \not\models \varphi$

We would like to ask $\boxed{\gamma = \mathsf{G}(? \Rightarrow \mathsf{F}publication)}$

γ is a temporal logic query [Chan 2000]

T.l.q.: . . . temporal logic queries

Given $\mathcal{S}$ and γ,

A solution to γ in $\mathcal{S}$ is a propositional formula f such that $\mathcal{S} \models \gamma(f)$ where $\gamma(f) = \gamma[f/?]$.

$paper\text{-}accepted$ is a solution to $\mathsf{G}(? \Rightarrow \mathsf{F}publication)$ in $\mathcal{S}$.

T.l.q.: that's interesting

Let $\varphi = \mathsf{G}(idea \Rightarrow \mathsf{F}publi.)$

Temporal query checking is a generalization of model checking:

- if $\mathcal{S} \models \varphi$ or $\mathcal{S} \not\models \varphi$: $\gamma = \mathsf{G}(? \Rightarrow \mathsf{F}publi.)$ is useful
- Example: if $\mathcal{S} \models \mathsf{G}(\top \Rightarrow \mathsf{F}publi.)$, we may have found a bug in our model !

See also [Bruns and Godefroid 2001]

T.l.q.: properties. . .

A query γ is said *positive* (resp. *negative*) if '?' appears under an even (resp. odd) number of negations. *Example:*

$\gamma = \mathsf{G}(? \Rightarrow \mathsf{F}publi.)$ is negative since

$? \Rightarrow \mathsf{F}publi. \equiv \neg ? \vee \mathsf{F}publi.$

From now on, we consider *positive* queries only

T.l.q.: . . . properties. . .

Lemma 1 (Monotonicity) *Let γ be a (positive) CTL* query and f and g two propositional formulae:*

$$f \supset g \quad \textit{entails} \quad \gamma(f) \supset \gamma(g)$$

T.l.q.: . . . properties

Lemma 1 shows that the set of solutions to γ in $\mathcal{S}$ is *closed* w.r.t. entailments (*upward-closed*).

A *minimal* solution to γ in $\mathcal{S}$ is a solution that is not entailed by any other solution.

The **QUERY_CHECKING** problem:

Input: $(\mathcal{S},\gamma)$

Output: The set of minimal solutions to γ in $\mathcal{S}$

T.l.q.: . . . properties. . .

Lemma 2 *$\mathcal{S}$: KS and γ: CTL* query:*

- $\mathcal{S} \models \gamma(\bot)$ *iff* $\mathcal{S} \models \gamma(f)$ *for all propositional formulae* f*,*
- $\mathcal{S} \models \gamma(\top)$ *iff* $\mathcal{S} \models \gamma(f)$ *for some propositional formula* f*.*

Existence of a solution to γ in $\mathcal{S}$ is reduced to a model-checking question: does $\mathcal{S} \models \gamma(\top)$?

Overview: the lattice of solutions

- Temporal logic queries
- The lattice of solutions
- Polynomial time algorithms
- Hardness results
- Conclusion

Lattice: preliminaries

- $\mathcal{P} = \{P_1, P_2, \ldots, P_m\}$.
- $\mathcal{V}$: set of valuations over $\mathcal{P}$.
- A propositional formula f denotes a set of valuations $[\![f]\!] \subseteq \mathcal{V}$: the valuations that satisfy f. $|f|_{\#} \stackrel{\text{def}}{=} [\![f]\!]$.

Lattice: the lattice $\mathcal{L}_2$ over $\mathcal{P} = \{P_1, P_2\}$

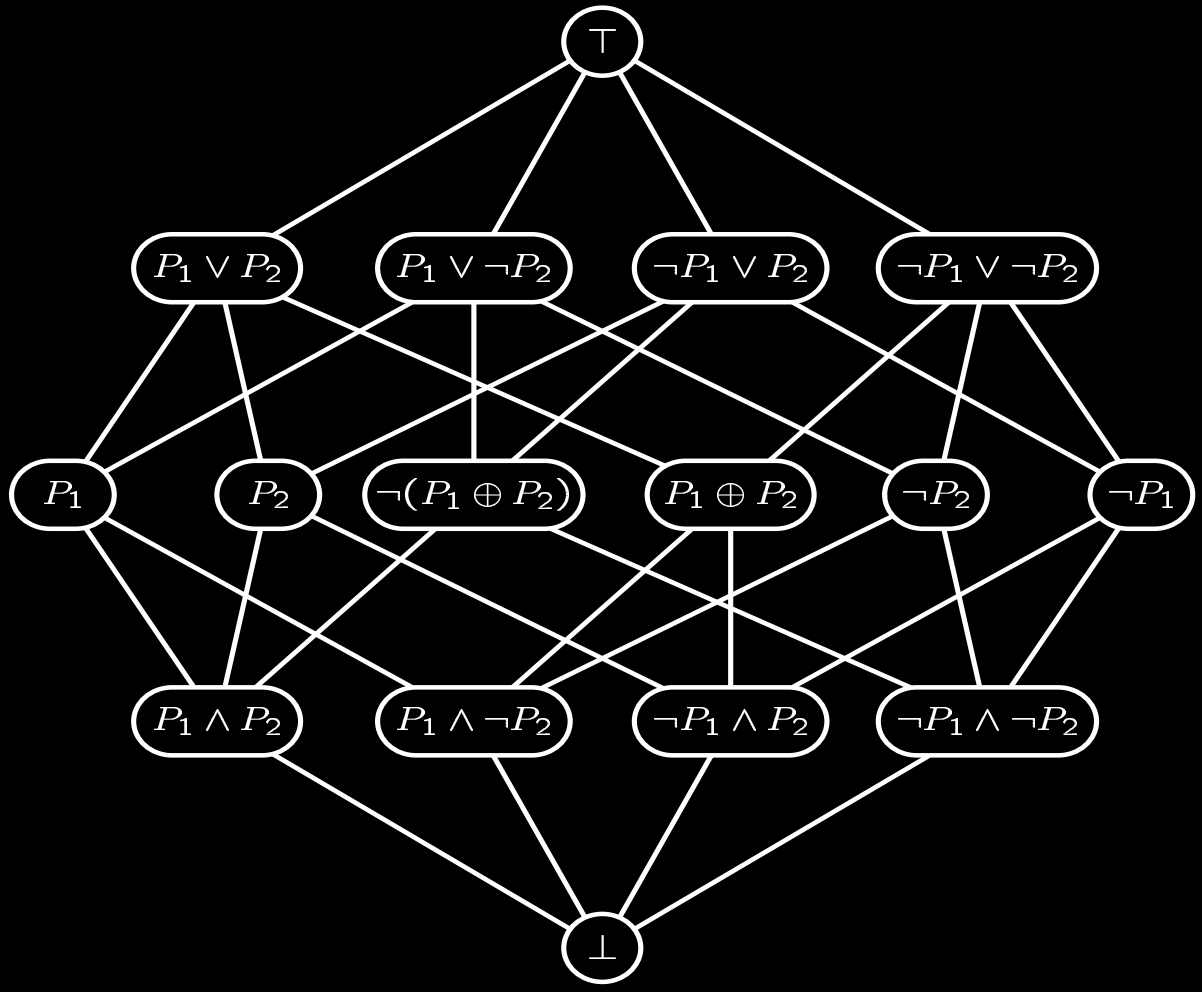
$|f|_{\#} = 4$:

$|f|_{\#} = 3$:

$|f|_{\#} = 2$:

$|f|_{\#} = 1$:

$|f|_{\#} = 0$:



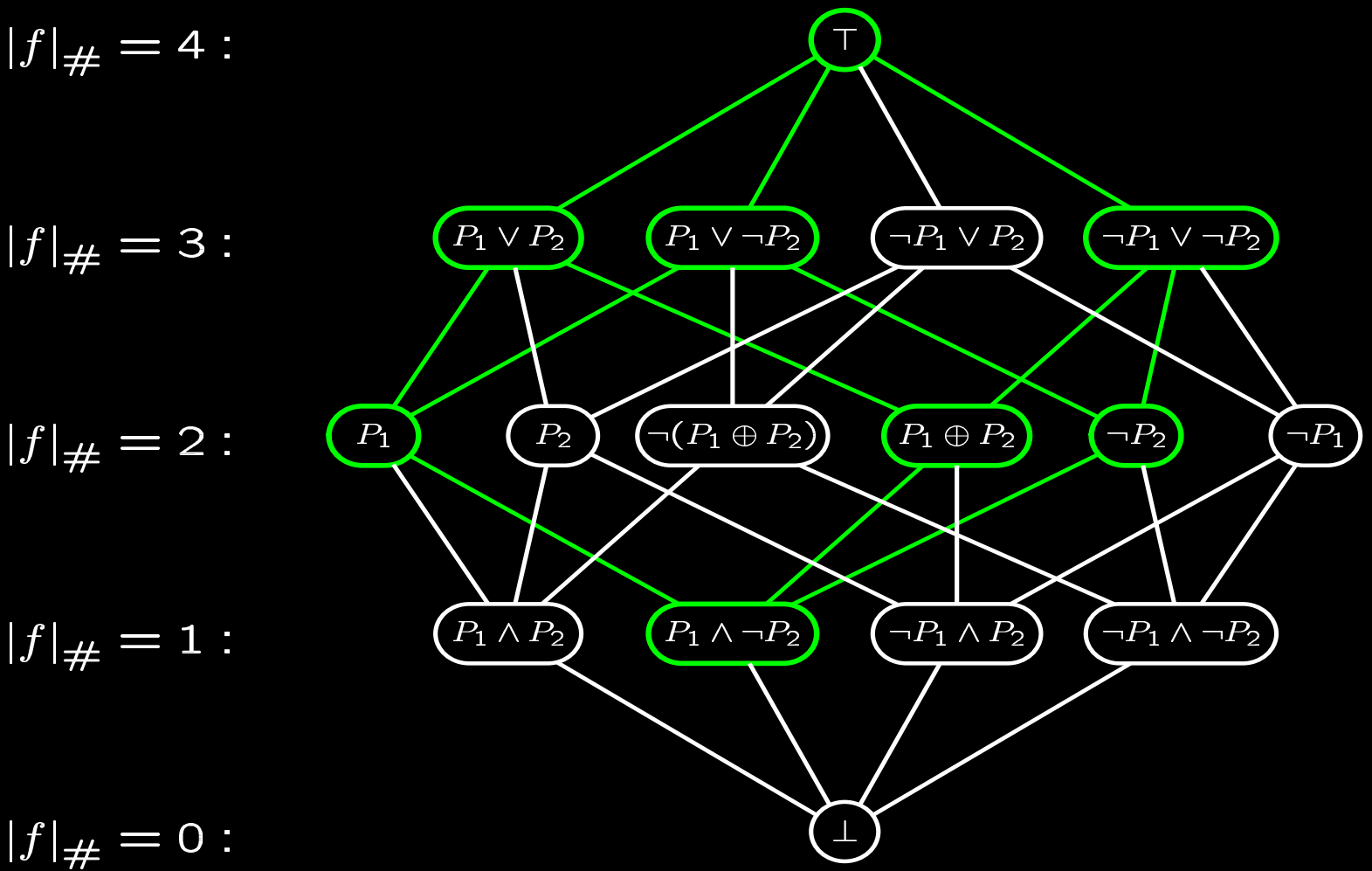
Lattice: some numbers

- $\mathcal{L}_m$ contains 2^{2^m} nodes (non equivalent propositional formulae).

- $\mathcal{L}_m$ can be sliced in $2^m + 1$ levels according to $|f|_{\#}$.

- The central level is the largest with $\binom{2^m}{2^{m-1}}$ propositional formulae.

Lattice: an upward-closed subset



Lattice: the number of minimal solutions

Lemma 3 *f: propositional formula. v: valuation. If v does not label any state of $\mathcal{S}$, then*

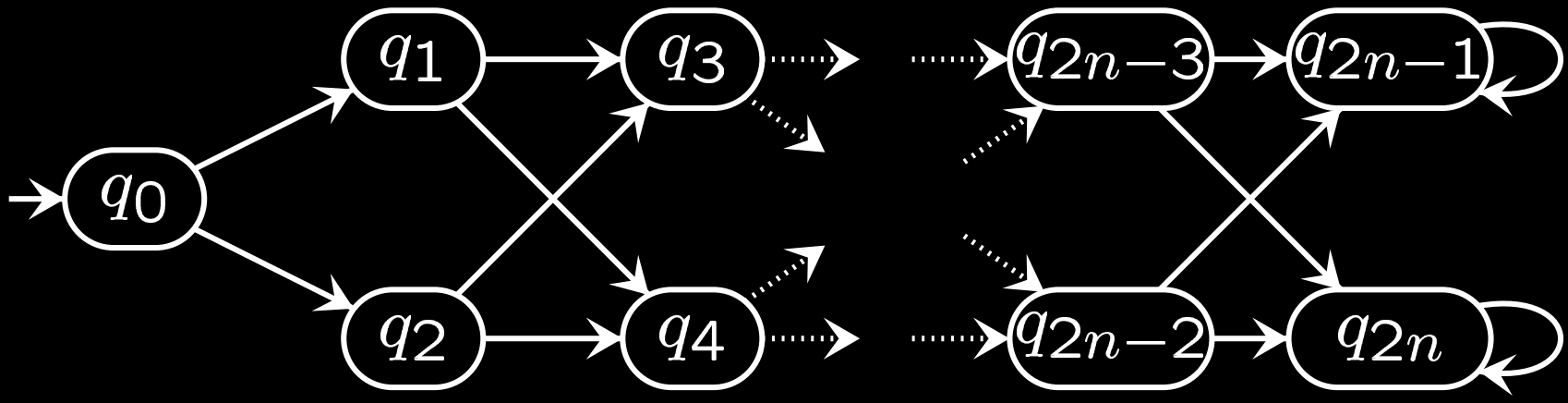
$\mathcal{S} \models \gamma(f \vee v)$ *iff* $\mathcal{S} \models \gamma(f)$.

Proposition 4 *γ: CTL* query. $\mathcal{S}$: KS with n nodes. Then γ has at most $\binom{n}{\lfloor n/2 \rfloor}$ minimal solutions, i.e. less than 2^n.*

Furthermore, $|f|_{\#} \leq n$ for any minimal solution f.

Lattice: many minimal solutions

$\mathcal{S}_n$:



Proposition 5 *In $\mathcal{S}_n$, the query $\gamma = EG\,?$ has 2^n minimal solutions.*

Lattice: relevant solutions

Important generalization of **QUERY_CHECKING**:

- Considering $\mathcal{RP} \subseteq \mathcal{P}$ leads to *relevant solutions*.
- A query has relevant solutions iff it has solutions (Lemma 2).
- $\mathcal{RP}$ remove the details.

Overview: polynomial time algorithms

- Temporal logic queries
- The lattice of solutions
- Polynomial time algorithms
- Hardness results
- Conclusion

P^{MC}: the MC oracle

We exhibit two temporal-query problems that can be reduced efficiently (polynomial-time Turing reduction) to the corresponding model checking problem for the underlying temporal logic.

These problems are in P^{MC}:

- Deciding minimality of a solution.
- Computing unique minimal solution.

P^{MC}: deciding minimality

Theorem 6 $\mathcal{S} = \langle \mathcal{Q}, q_{\mathsf{init}}, \rightarrow, l \rangle$.

γ: *query.*

f: *propositional formula.*

Deciding whether f *is* a minimal solution *to* γ *in* $\mathcal{S}$, *is in* P^{MC} *when* f *is given in disjunctive normal form, or as an OBDD.*

P^{MC}: deciding minimality: illustration

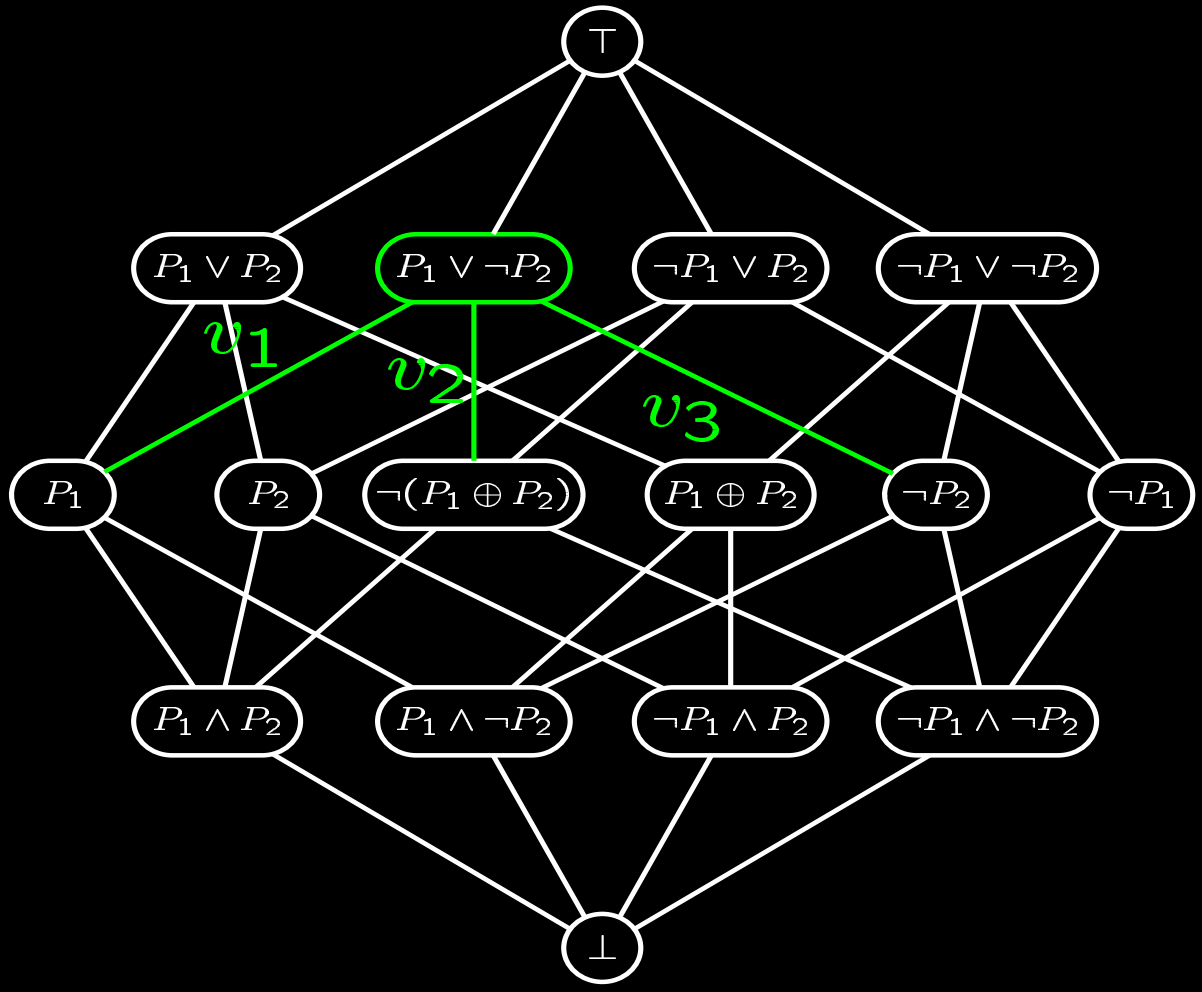
$|f|_{\#} = 4$:

$|f|_{\#} = 3$:

$|f|_{\#} = 2$:

$|f|_{\#} = 1$:

$|f|_{\#} = 0$:



P^{MC}: deciding minimality: proof

1. if $\mathcal{S} \not\models \gamma(f)$, then f is not a solution
2. if $|f|_{\#} > |\mathcal{Q}|$, then f is not minimal (prop. 4)
3. enumerate $[\![f]\!]$ as $\{v_1, \ldots, v_k\}$. f is minimal iff $\mathcal{S} \not\models \gamma(f \wedge \neg v_i)$ for $i = 1, \ldots, k$

We do at most $1 + |\mathcal{Q}|$ invocations to a model-checking algorithm

P^{MC}: computing unique minimal solution

Theorem 7 $\mathcal{S} = \langle \mathcal{Q}, q_{\mathsf{init}}, \rightarrow, l \rangle$.

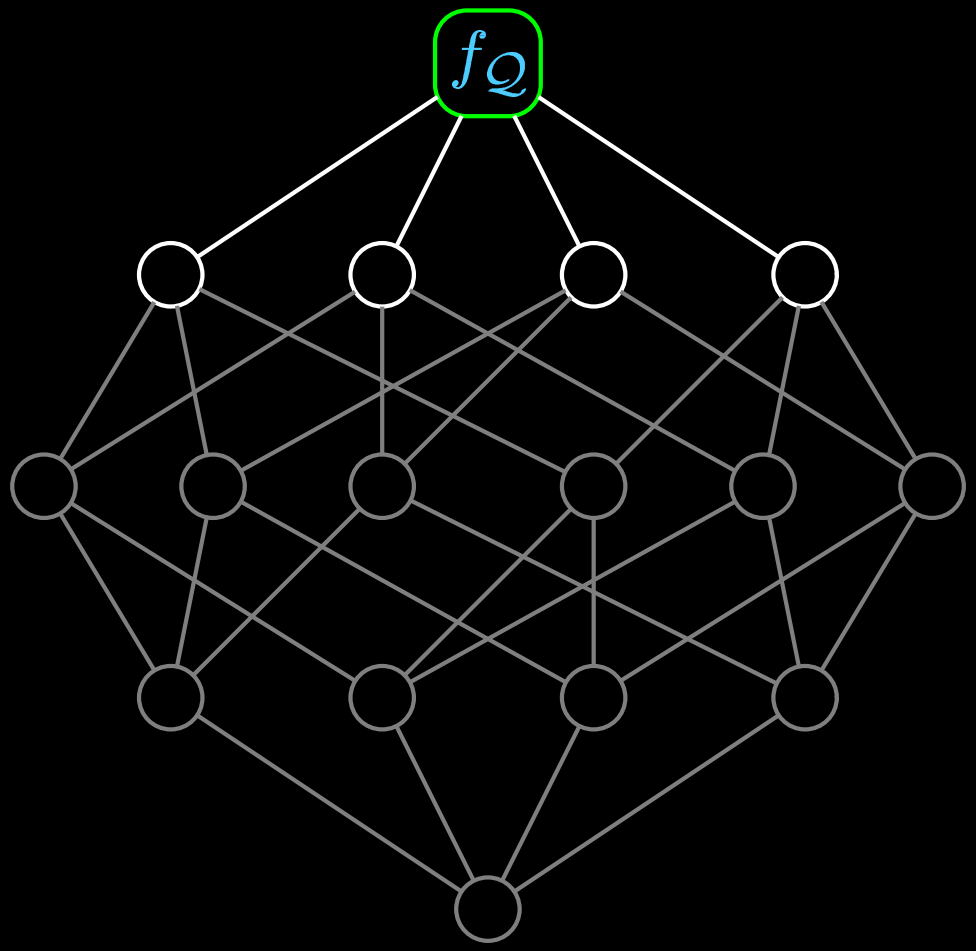
γ: *query.*

Deciding whether γ *admits a unique minimal solution in* $\mathcal{S}$ *(and computing that solution) is in* P^{MC}.

P^{MC}: comp. uni. min. sol.: proof...

1. let $\mathcal{V}_{\mathcal{Q}}$ be the set $\{v_q \mid q \in \mathcal{Q}\}$ of valuations labeling a state of $\mathcal{S}$. Define

$$f_{\mathcal{Q}} = \bigvee_{v \in \mathcal{V}_{\mathcal{Q}}} v = \bigvee_{q \in \mathcal{Q}} v_q$$

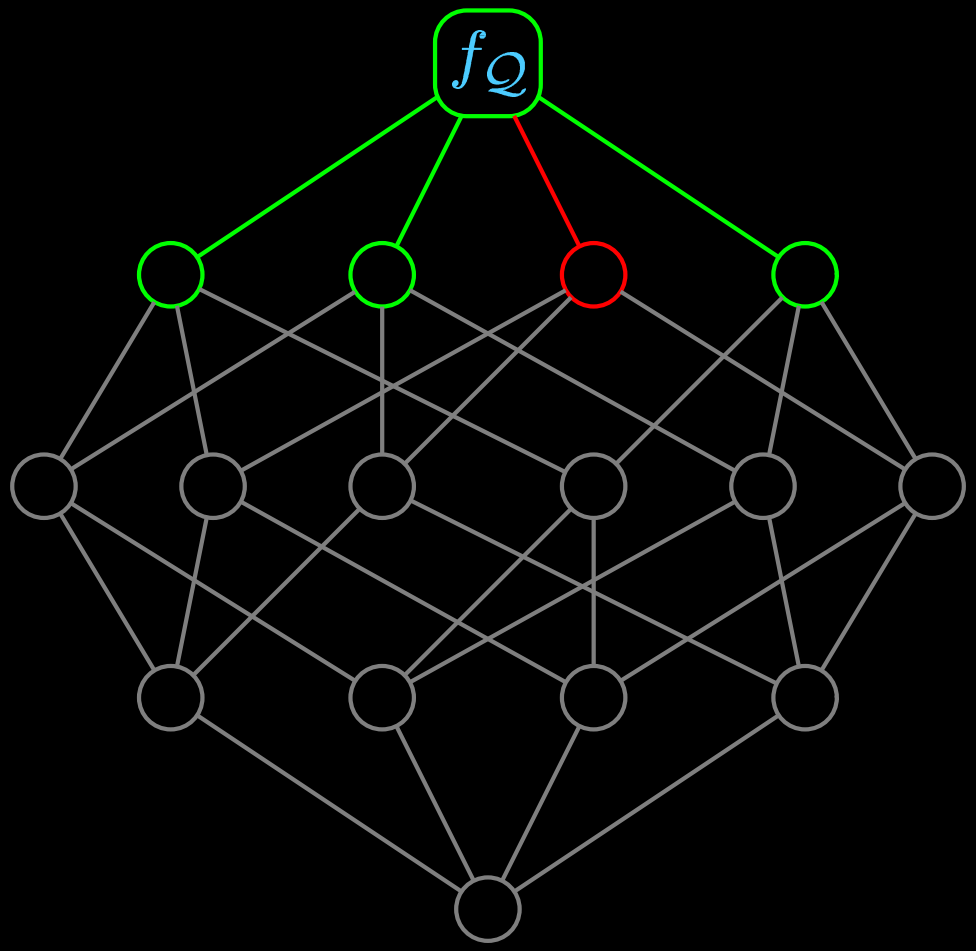


P^{MC}: comp. uni. min. sol.: proof...

2. let $\mathcal{V}'$ be the set

$\{v \in \mathcal{V}_{\mathcal{Q}} \mid$

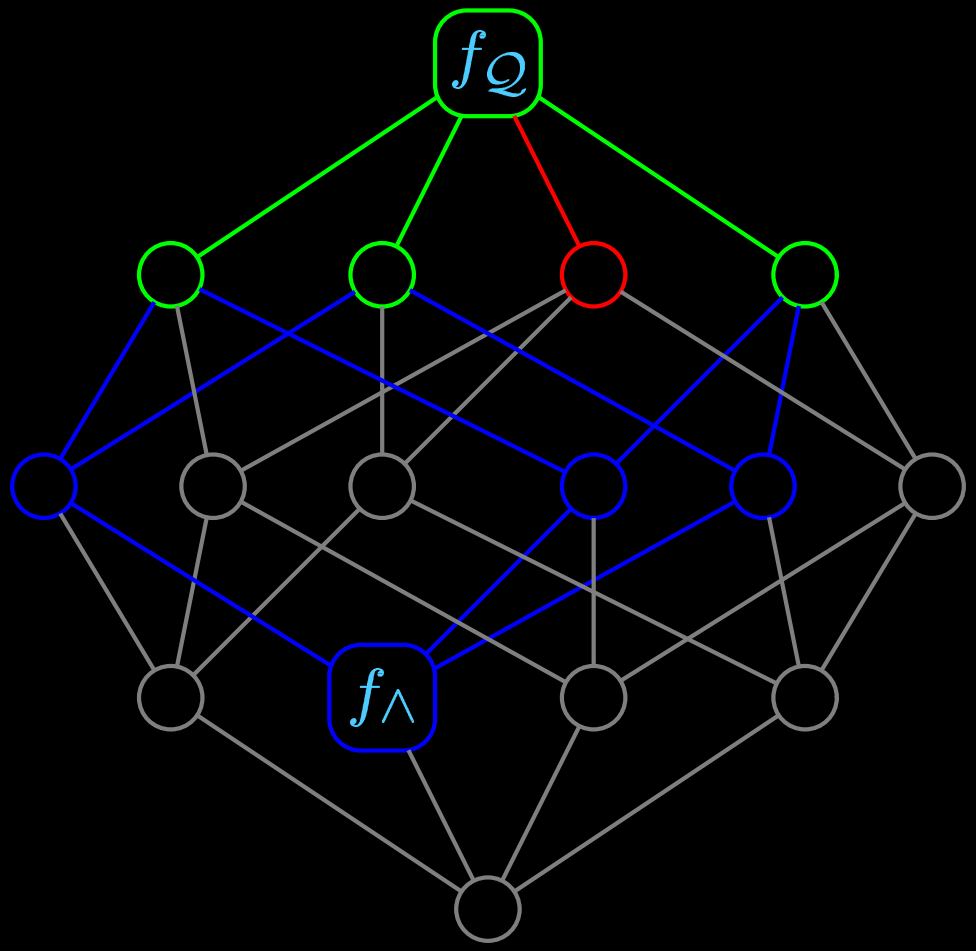
$\mathcal{S} \models \gamma(f_{\mathcal{Q}} \wedge \neg v)\}$



P^{MC}: comp. uni. min. sol.: proof...

3. Define

$$f_\wedge = \bigvee_{v \in \mathcal{V}_\mathcal{Q} \setminus \mathcal{V}'} v$$



P^{MC}: comp. uni. min. sol.: . . . proof

4. if $\mathcal{S} \models \gamma(f_\wedge)$ then $f_\wedge$ is the unique minimal solution. Otherwise, there is no solution, or the minimal solution is not unique.

We do at most $1 + |\mathcal{Q}|$ invocations to a model-checking algorithm

P^{MC}: many minimal solutions

Given $f_1, \ldots, f_k$ minimal solutions to γ in $\mathcal{S}$

Do they form the complete set of minimal solutions ?: $O(|\mathcal{Q}|^k + |\mathcal{Q}|)$ invocations

If it is not the case, one can compute (at the same time) an additional f_{k+1} minimal solution.

Overview: hardness results

- Temporal logic queries
- The lattice of solutions
- Polynomial time algorithms
- Hardness results
- Conclusion

Hardness results

Two problems on the number of minimal solutions.

COUNT_SOLS

Input: $\mathcal{S}$, γ

Output: the number of minimal solutions to γ in $\mathcal{S}$

MANY_SOLS

Input: $\mathcal{S}$, γ, and an integer k in unary

Output: *yes* if there are at least k minimal solutions to γ in $\mathcal{S}$

Hardness results

Theorem 8 *Considering CTL queries,*

COUNT_SOLS *is #P-complete.*

MANY_SOLS *is NP-complete.*

Hardness already appears with the fixed query EG ?

Proof: The easy part is that **MANY_SOLS** is in NP and **COUNT_SOLS** is in #P

Hardness of COUNT_SOLS

We reduce #SAT to **COUNT_SOLS**.

Let $\mathcal{I}$ be a SAT instance with n boolean variables in CNF.

$$\mathcal{I}: \quad x_2 \vee x_3 \ \wedge \ x_1 \vee x_2 \vee \overline{x_3} \ \wedge \ \overline{x_2} \vee \overline{x_3}$$

With $\mathcal{I}$, we associate the Kripke structure $\mathcal{S}_{\mathcal{I}}$

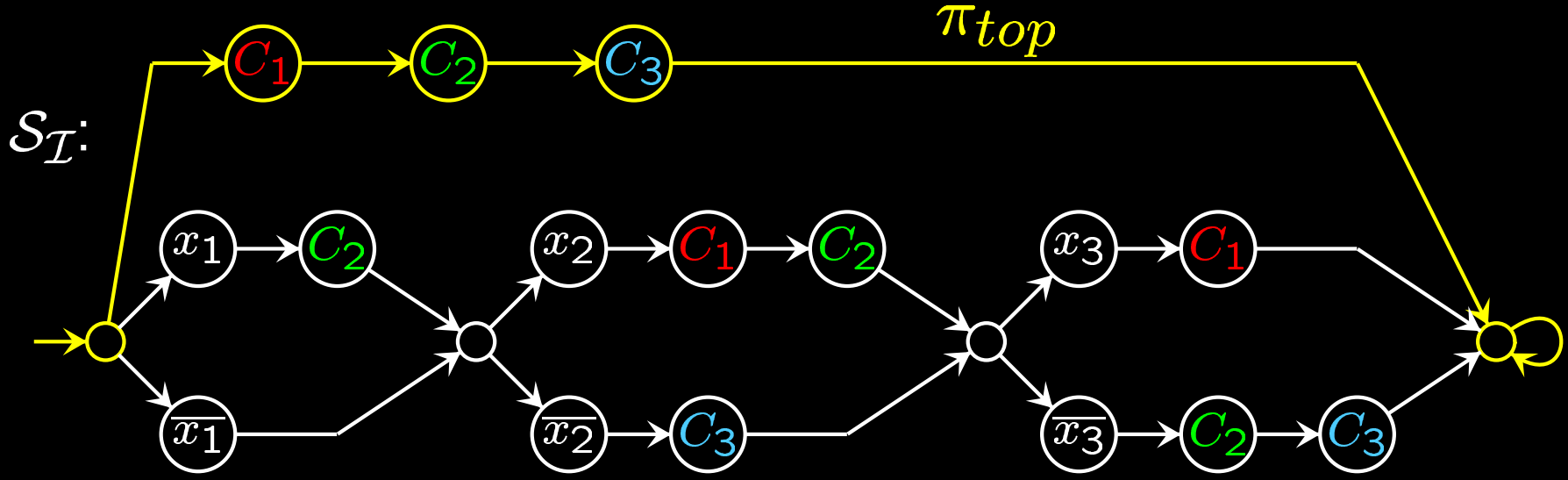
With a path π, we associate

$$f_\pi = \bigvee_{q \in \pi} v_q$$

Hardness of COUNT_SOLS

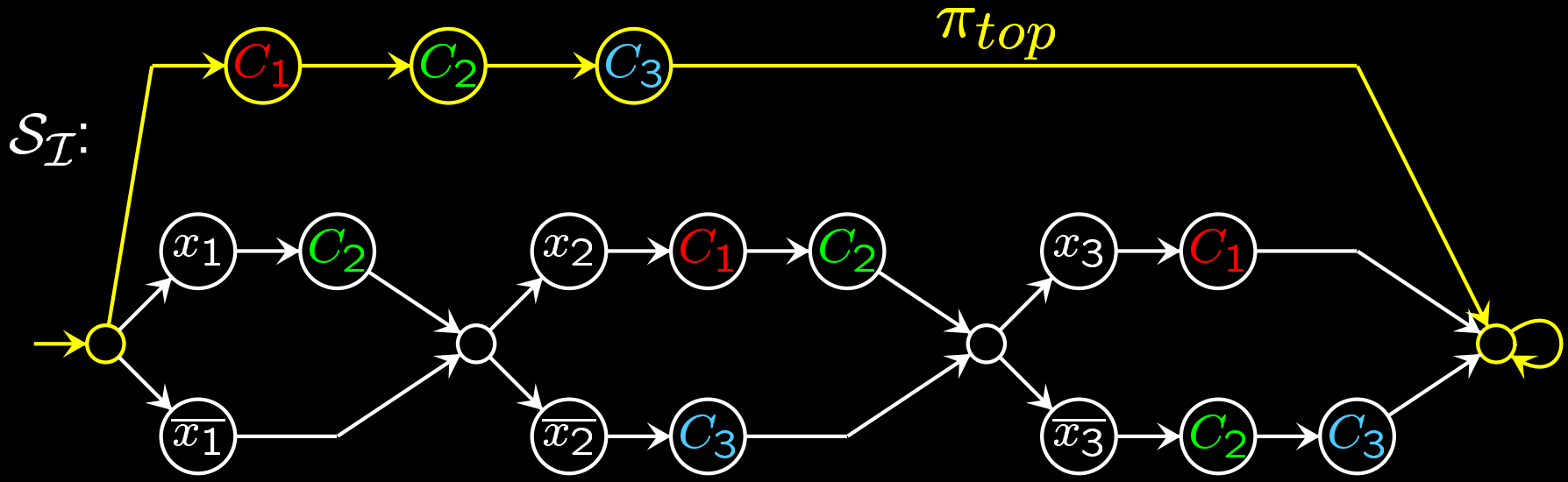
$\mathcal{I}$: $\underbrace{x_2 \vee x_3}_{C_1} \wedge \underbrace{x_1 \vee x_2 \vee \overline{x_3}}_{C_2} \wedge \underbrace{\overline{x_2} \vee \overline{x_3}}_{C_3}$

$\gamma =$ EG?



Hardness of COUNT_SOLS

$\mathcal{I}$: $\underbrace{x_2 \vee x_3}_{C_1} \wedge \underbrace{x_1 \vee x_2 \vee \overline{x_3}}_{C_2} \wedge \underbrace{\overline{x_2} \vee \overline{x_3}}_{C_3}$ $\gamma = \text{EG?}$



There are $2^n + 1 - k$ min. sols. where k is the solution to #SAT (for $\mathcal{I}$, $n = 3$ and $k = 3$)

Hardness of MANY_SOLS

We reduce SAT to MANY_SOLS.

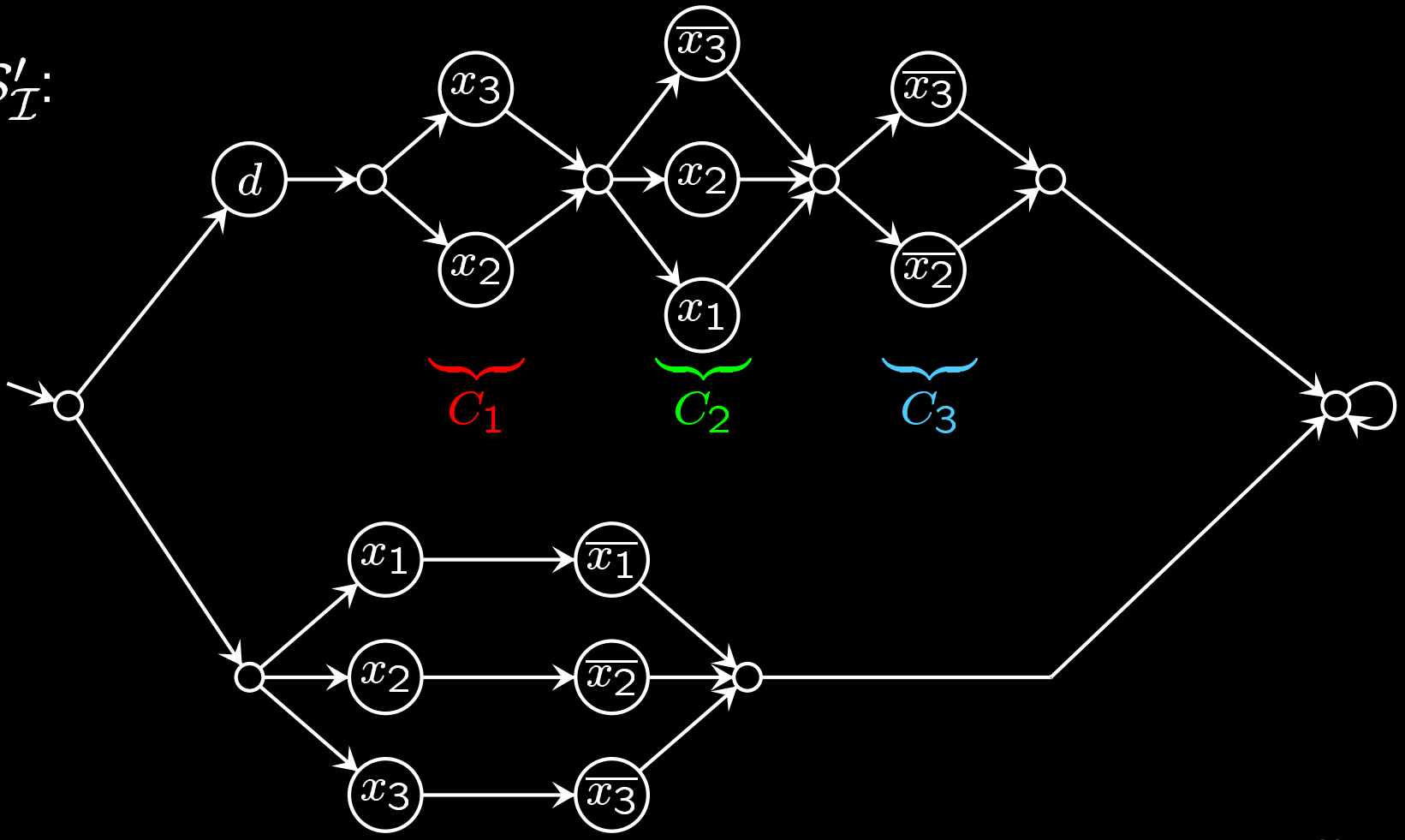
We illustrate on the earlier example.

With $\mathcal{I}$, we associate the Kripke structure $\mathcal{S}'_{\mathcal{I}}$

Hardness of MANY_SOLS

$\mathcal{I}$: $\underbrace{x_2 \vee x_3}_{C_1} \wedge \underbrace{x_1 \vee x_2 \vee \overline{x_3}}_{C_2} \wedge \underbrace{\overline{x_2} \vee \overline{x_3}}_{C_3}$ $\gamma =$ EG?

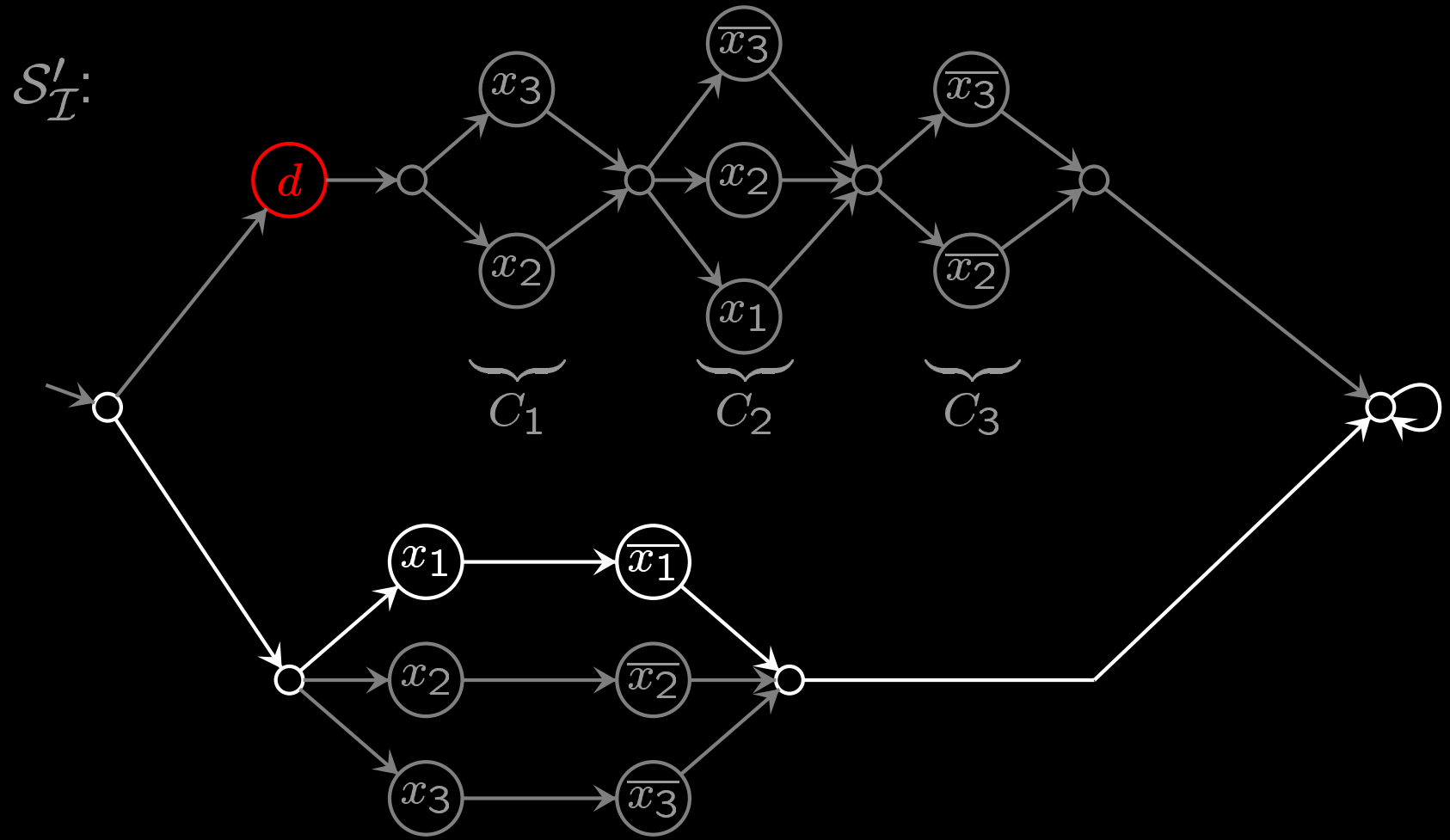
$\mathcal{S}'_{\mathcal{I}}$:



Hardness of MANY_SOLS

$\mathcal{I}$: $\underbrace{x_2 \vee x_3}_{C_1} \wedge \underbrace{x_1 \vee x_2 \vee \overline{x_3}}_{C_2} \wedge \underbrace{\overline{x_2} \vee \overline{x_3}}_{C_3}$

$\gamma =$ EG?



Overview: conclusion

- Temporal logic queries
- The lattice of solutions
- Polynomial time algorithms
- Hardness results
- Conclusion

Conclusion

- computing the set of min. sols. to any γ over any $\mathcal{S}$ is intractable in general

- it is easier to ask for a few min. sols, or the existence of a unique one

- no implementation yet, but that would make sense and has already proved useful

Thank you

Any questions ?

Acknowledgments

We thank Stéphane Demri for help on **MANY_SOLS** and Paul Gastin for the GasTeX package for LaTeX

References

[Chan 2000] *Temporal-logic queries*, William Chan, CAV'2000

[Bruns and Godefroid 2001] *Temporal logic query checking*, G. Bruns and P. Godefroid, LICS'2001